APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/530,375	04/06/2005	Carlos Antonio Alba Pinto	NL 020979	4877

24737   7590   07/09/2007

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER
OTTO, ALAN

ART UNIT	PAPER NUMBER
2187	

MAIL DATE	DELIVERY MODE
07/09/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/530,375	ALBA PINTO ET AL.
	Examiner	Art Unit	
	Alan M. Otto	2187	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>06 April 2005</u>.
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-14</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) <u>1-14</u> is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on <u>06 April 2005</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a) ☒ All b) ☐ Some * c) ☐ None of:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____.
      3. ☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date <u>9/12/2005</u>.
4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detailed Action

The instant application having Application No. 10/530375 has a total of 14 claims pending in the application, there are 2 independent claim and 12 dependent claims, all of which are ready for examination by the examiner.

INFORMATION CONCERNING OATH/DECLARATION

Oath/Declaration

1.   The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

> The oath or declaration is defective because: It does not state that the person making the oath or declaration acknowledges the duty to disclose to the Office all information known to the person to be material to patentability as defined in 37 CFR 1.56.

INFORMATION CONCERNING DRAWINGS

Drawings

2.   The applicant's drawings submitted April 6th, 2005 are acceptable for examination purposes.

ACKNOWLEDGEMENT OF REFERENCES CITED BY APPLICANT

Information Disclosure Statement

3.   As required by **M.P.E.P. 609 (C)**, the applicant's submission of the Information Disclosure Statement dated September 12th, 2005 is acknowledged by the examiner

and the cited references have been considered in the examination of the claims now

pending. As required by **M.P.E.P. 609 C(2)**, a copy of the PTOL-1449 initialed and

dated by the examiner is attached to the instant office action.

REJECTIONS NOT BASED ON PRIOR ART

a. DEFICIENCIES IN THE CLAIMED SUBJECT MATTER

Claim Objections

4.   Claims 2-5, 8, and 14 are objected to because of the following informalities:

In claim 2, lines 1-2, "the instruction address modification unit" is mentioned, but

has not been previously defined. Examiner suggests changing "the instruction address

modification unit" to –the instruction address modification circuit–. Claims 3-5, 8, and 14

are dependent on claim 2.

Appropriate correction is required.

REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(e) the invention was described in (1) an application for patent, published under section 122(b), by
another filed in the United States before the invention by the applicant for patent or (2) a patent
granted on an application for patent by another filed in the United States before the invention by the
applicant for patent, except that an international application filed under the treaty defined in section
351(a) shall have the effects for purposes of this subsection of an application filed in the United States
only if the international application designated the United States and was published under Article 21(2)
of such treaty in the English language.

5.　　Claims 1-2, 4, 8-9, 11-12, and 14 are rejected under 35 U.S.C. 102(e) as being

anticipated by Schlansker (U.S. Patent No. 7,024,538), herein after referred to as

Schlansker'538.

6.　　Referring to claim 1, Schlansker'538 discloses as claimed, a data processing

apparatus, the apparatus comprising an instruction addressing unit (**see fig. 4,**

controller 55, which controls addressing by using lookup table 57, main memory

56, and individual instruction memories); an instruction memory system (**see fig. 4,**

regarding main memory 56, and processor specific instruction memories such as

instruction memory 53) arranged to output an instruction word (**see col. 5, lines 1-4,**

regarding the main memory holding a super instruction), capable of containing a

plurality of instructions (**see fig. 3, regarding instruction words C1-C5**), in response to

an instruction address from the instruction addressing unit (**see col. 5, lines 43-55,**

regarding the target address being used from the controller to find local

instruction memory addresses), the instruction memory system comprising a plurality

of memory units (**see fig. 4, local instruction memory 53, where each processing**

unit has a local instruction memory), arranged to output respective parts of the

instruction word in parallel (**see col. 5, lines 43-55, regarding the functional units**

obtaining instructions for processing after a branch, and resuming execution.

Also see col. 5, lines 37-42, regarding the processing units stepping through

instructions in synchronization); an instruction execution unit, comprising a plurality

of functional units (**see col. 5, lines 27-30, regarding a processor connected to a**

plurality of processing sections), each capable of executing a respective instruction

from the instruction word in parallel with execution of other instructions from the

instruction word by other ones of the functional units (**see col. 3, lines 43-46,**

regarding sets of instructions from a super instruction each being executed by

each processor on each cycle, thus operating in parallel); an instruction address

modification circuit arranged to modify translation of the instruction address into a

physical address for a particular one of the memory units relative to other ones of the

memory units during program execution (**see col. 6, lines 35-48, regarding where the**

memories are caches, and a tag array is used to translate target addresses to

local addresses for each corresponding functional unit for the cache of that

functional unit).

7.   As to claim 2, Schlansker'538 discloses as claimed, a data processing apparatus

according to Claim 1, wherein the instruction address modification unit is arranged to

modify the translation under control of a modification update instruction from the

instruction word during program execution (**see fig. 3, regarding a super instruction,**

where the header stipulates how the instruction will be modified and broken

down into several different parts or tuples, as stipulated in col. 5, lines 1-25. If

there is no tuple for a functional unit, then it is interpreted as a sequence of

NOOPs. Therefore, the super instruction header contains information of how the

instruction will be modified).

8.   As to claim 4, Schlansker'538 discloses as claimed, a data processing apparatus

according to Claim 2, wherein the particular one of the memory units is arranged to

supply instructions exclusively to a group that contains a subset of the functional units **(see fig. 4, where functional unit 52 receives instructions from instruction memory 53, and functional unit 52 is a group that is a subset of a number of functional units, as shown by other processor section 51**), the functional units comprising a modification update functional unit outside the group constructed to execute the modification update instruction **(see fig. 4, regarding the controller 55, which could qualify as a functional unit, and is outside the group of other functional units as shown in fig. 4).**

9.   As to claim 8, Schlansker'538 discloses as claimed, a data processing apparatus according to Claim 2, programmed to use said modification update instruction **(see col. 5, lines 1-25, where an instruction word is broken into tuples)** to selectively output, dependent on a data dependent condition **(see col. 5, lines 1-25, where depending on if the functional unit is idle, data might be sent. Otherwise NOOPs are given.),** a first or a second block of one or more instructions from said particular one of the memory units **(see fig. 4, lookup table 57, where an address for a functional unit 1 would link and point to instructions. See fig. 3, where blocks of instructions are shown in the instruction word),** while the instruction address progresses so that at least part of the memory units output one or more instructions from a third block of instructions **(see fig. 4, lookup table 57, where an address for a functional unit n would link to different instructions than the first. Also see fig. 3, where C5 represents different instructions from C1)** as part of the instruction word or words in combination with instructions from said first or second block **(see col. 5, lines 1-25,**

where each of these parts of instructions are tuples, and part of one instruction word).

9.   As to claim 9, Schlansker'538 discloses as claimed, a data processing apparatus according to Claim 1, wherein a first number of addressable instruction addresses of the particular one of the memory units differs from a second number of addressable instruction addresses of at least one of the memory units (**see fig. 4, where lookup table 57 gives separate addresses for each functional unit based on a common block address**).

10.   As to claim 11, Schlansker'538 discloses as claimed, a method of executing a program of instruction words with a data processing apparatus that comprises a plurality of functional units (**see col. 5, lines 27-30, regarding a processor connected to a plurality of processing sections**) capable of executing a plurality of instructions from each instruction word in parallel (**see col. 3, lines 43-46, regarding sets of instructions from a super instruction each being executed by each processor on each cycle, thus operating in parallel**), wherein the instructions from each of at least some of the instruction words are fetched from respective memory units in parallel (**see col. 7, lines 32-36, regarding parallel lookups for operations for each functional unit, and then each unit begins interpreting the stream of operations**), the method comprising addressing the instruction word with an instruction address that is common for the functional units (**see col. 3, lines 42-47, regarding broadcasting the address of the super instruction to all functional units during a branch**); using a modifiable translation of the instruction address into a physical address for a particular one of the

memory units (**see col. 6, lines 35-48, regarding where the memories are caches, and a tag array is used to translate target addresses to local addresses for each corresponding functional unit for the cache of that functional unit**) to select dependent on program execution which instructions from the memory units will be combined into the instruction word in response to the instruction address (**see col. 5, lines 1-25, regarding tuples from the super instruction going to individual memory units. If a functional unit remains idle, which could be dependent on program execution, then NOOPs would be used as a sequence of operations for a particular functional unit. Otherwise, the tuples from the super instruction can be used**).

11.   As to claim 12, Schlansker'538 discloses as claimed, a method of executing a program of instruction words according to Claim 11, wherein the modifiable translation is selected under control of a modification update instruction in the program (**see fig. 3, regarding a super instruction, where the header stipulates how the instruction will be modified and broken down into several different parts or tuples, as stipulated in col. 5, lines 1-25. If there is no tuple for a functional unit, then it is interpreted as a sequence of NOOPs. Therefore, the super instruction header contains information of how the instruction will be modified**).

12.   As to claim 14, Schlansker'538 discloses as claimed, a computer program product comprising instruction words (**see col. 5, lines 1-2, regarding the main memory containing super instructions**), each for execution in one or more respective instruction cycles by a data processing apparatus (**see col. 5, lines 39-41, regarding**

the processing sections stepping through the instructions on each cycle)

according to Claim 2, the instruction words comprising at least one modification update

instruction **(see fig. 3, regarding a super instruction, where the header stipulates**

how the instruction will be modified and broken down into several different parts

or tuples, as stipulated in col. 5, lines 1-25. If there is no tuple for a functional

unit, then it is interpreted as a sequence of NOOPs. Therefore, the super

instruction header contains information of how the instruction will be modified)

for causing the data processing apparatus to execute, upon addressing a first one of the

instruction words, a combination of instructions from the first one of the instruction word

(see fig. 3, where the instructions to functional unit 1, or C1 would constitute first

instruction words) with further instructions from outside the first instruction word **(see**

fig. 3, where instructions to functional unit 2, or C2 would constitute instructions

from outside the first instruction word), following execution of the modification

update word **(see col. 5, lines 1-25, regarding first interpreting the super**

instruction block, including separating tuples, and inserting NOOPs into idle

functional units).

Claim Rejections - 35 USC 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

13.   Claims 3 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Schlansker'538, in view of Moroney et al. (U.S. Patent Application Publication No.

2003/0145116), herein referred to as Moroney et al.'116.

14.   As to claim 3, Schlansker'538 discloses as claimed, a data processing apparatus

according to Claim 2, wherein the particular one of the memory units is arranged to

supply instructions exclusively to a group that contains a subset of the functional units

(see fig. 4, where functional unit 52 receives instructions from instruction

memory 53, and functional unit 52 is a group that is a subset of a number of

functional units, as shown by other processor section 51).

Schlansker'538 discloses the claimed invention except for the group containing a

modification update functional unit constructed to execute the modification update

instruction.

However, Moroney et al.'116 discloses the group containing a modification

update functional unit constructed to execute the modification update instruction **(see**

para. 76, lines 1-4, regarding an ALU which includes a translator block. Also see

para. 82, lines 8-13, regarding the translator block handling jumps based on case

statements).

Schlansker'538 and Moroney et al.'116 are analogous art because they are from

the same field of endeavor of VLIW systems **(see Schlansker'538, col. 4, lines 30-34,**

regarding a VLIW processor. See Moroney et al.'116, para. 5, lines 4-12, regarding

a system that executes very long instruction words in different functional units in

parallel).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Schlansker'538 to comprise the group containing a

modification update functional unit constructed to execute the modification update

instruction, as taught by Moroney et al.'116, in order to enable the functional units to

operate in parallel with each other (**see para. 5, lines 6-12, regarding the VLIW**

system having different functional blocks which enable parallel operation. By

having the modification update unit in a functional block, it enables parallel

operation, and handles instructions in a similar manner to other functional units).

15.   As to claim 5, Schlansker'538 discloses the claimed invention except for a data

processing apparatus according to Claim 2 wherein the modification update instruction

is a conditional instruction, the modification update being executed dependent on

fulfillment of a condition specified in the modification update instruction.

However, Moroney et al.'116 discloses where the modification update instruction

is a conditional instruction (**see para. 82, lines 8-13, regarding the translator block**

handling jumps based on case statements), the modification update being executed

dependent on fulfillment of a condition specified in the modification update instruction

(**see para. 83, where a case statement is described. If a comparison is true, then**

the jump occurs, and the instruction address is changed, otherwise the jump

does not occur).

Schlansker'538 and Moroney et al.'116 are analogous art because they are from

the same field of endeavor of VLIW systems (**see Schlansker'538, col. 4, lines 30-34,**

regarding a VLIW processor. See Moroney et al.'116, para. 5, lines 4-12, regarding

a system that executes very long instruction words in different functional units in parallel).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Schlansker'538 to comprise where the modification update instruction is a conditional instruction, the modification update being executed dependent on fulfillment of a condition specified in the modification update instruction, as taught by Moroney et al.'116, in order to allow for conditional branching (see para. 46, lines 6-10, regarding the case statement allowing for conditional branching, and being able to maintain state information more effectively).

16.   Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Schlansker'538, in view of the examiner taking official notice.

17.   As to claim 6, Schlansker'538 discloses as claimed; a data processing apparatus according to Claim 1, wherein the instruction address modification circuit is arranged for instruction address and memory unit dependent address translation (see fig. 4, lookup table 57, which shows that the translation is dependent on a memory unit corresponding to a functional unit, and is found by giving a block address), so that a first and a second instruction address are to mutually different physical addresses for one or more memory units other than the particular one of the memory traits (see fig. 4, lookup table 57, where a given block address is given, and addresses corresponding to functional units are found. Also see col. 5, lines 47-55).

Schlansker'538 discloses the claimed invention except where a first and a

second instruction address are translated to a same physical address for the particular

one of the memory units.

However, it is well known in the art that virtual address to physical address

translation enables two different virtual addresses to share the same physical address

when translated.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Schlansker'538 to comprise where a first and a second

instruction address are translated to a same physical address for the particular one of

the memory units, in order to use virtual addressing, and therefore map non-contiguous

memory through the use of contiguous virtual addresses, or to allow for easier mapping

across multiple memories through virtual addressing.

18.   Claims 7 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Schlansker'538, in view of Fleck et al. (U.S. Patent No. 6,076,159), herein referred to as

Fleck et al.'159.

19.   As to claim 7, Schlansker'538 discloses the claimed invention except for a data

processing apparatus according to Claim 1, programmed to use repeated modification

of said translation to repeatedly output one or more instructions making up a loop of

instructions, while the instruction address progresses so that the instructions from the

loop are combined in the instruction words with progressive instructions that are not

repeated during at least part of the repetitions of supply of instructions from the loop.

However, Fleck et al.'159 discloses a data processing apparatus programmed to use repeated modification of said translation to repeatedly output one or more instructions making up a loop of instructions (**see col. 7, lines 57-67, and col. 8, lines 1-9, regarding a third pipeline which executes a loop. Also see fig. 1, regarding loop pipeline 12. More detail is shown in fig. 3, which shows the loop execution unit 4**), while the instruction address progresses so that the instructions from the loop are combined in the instruction words with progressive instructions (**see col. 7, lines 1-10, regarding the first and second pipelines which are the data pipeline and the address pipeline. Also see fig. 1, regarding the two major pipelines 10 and 11**) that are not repeated during at least part of the repetitions of supply of instructions from the loop (**see col. 7, lines 60-62, where a loop operation is performed in parallel with an integer and a load/store operation**).

Schlansker'538 and Fleck et al.'159 are analogous art because they are from the same field of endeavor of parallel execution in multi-processor systems (**see Schlansker'538, col. 4, lines 30-34, regarding a VLIW processor with multiple functional units. See Fleck et al.'159, col. 1, lines 59-67, regarding a data processor executing with parallel pipelines**).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Schlansker'538 to comprise where the modification update instruction is a conditional instruction, the modification update being executed dependent on fulfillment of a condition specified in the modification update instruction, as taught by Fleck et al.'159, in order to allow for one the execution of a loop and

additional code in parallel, and therefore enable highly efficient parallel execution (**see col. 8, lines 8-12, and col. 10, lines 22-26, regarding how the parallel execution of three pipelines is useful for DSP related instructions**).

20.   As to claim 13, Schlansker'538 discloses the claimed invention except for a method of executing a program of instruction words according to Claim 11, comprising modifiable translation to repeatedly fetch instructions from a loop from repeated physical addresses in a particular one of the memory units in response to progressive instruction addresses, so that the instructions from at least part of repetitions of the loop are combined in the instruction words with progressively different instructions memory units other the than the particular one of the memory unit.

However, Fleck et al.'159 discloses a method of executing a program of instruction words according to Claim 11, comprising modifiable translation to repeatedly fetch instructions from a loop from repeated physical addresses (**see col. 7, lines 64-65, and col. 8, lines 5-7, regarding an instruction being fetched the first time, but loaded into a buffer for subsequent loop operations, which would mean in subsequent loop operations, the address is being fetched from a repeated address**) in a particular one of the memory units in response to progressive instruction addresses (**see col. 2, lines 8-29, regarding the steps of executing an instruction, where the instructions would be executed progressively**), so that the instructions from at least part of repetitions of the loop are combined in the instruction words with progressively different instructions memory units other the than the particular one of the

memory unit (**see col. 7, lines 60-62, where a loop operation is performed in parallel with an integer and a load/store operation**).

Schlansker'538 and Fleck et al.'159 are analogous art because they are from the same field of endeavor of parallel execution in multi-processor systems (**see Schlansker'538, col. 4, lines 30-34, regarding a VLIW processor with multiple functional units. See Fleck et al.'159, col. 1, lines 59-67, regarding a data processor executing with parallel pipelines**).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Schlansker'538 to comprise a method of executing a program of instruction words according to Claim 11, comprising modifiable translation to repeatedly fetch instructions from a loop from repeated physical addresses in a particular one of the memory units in response to progressive instruction addresses, so that the instructions from at least part of repetitions of the loop are combined in the instruction words with progressively different instructions memory units other the than the particular one of the memory unit, as taught by Fleck et al.'159, in order to allow for one the execution of a loop and additional code in parallel, and therefore enable highly efficient parallel execution (**see col. 8, lines 8-12, and col. 10, lines 22-26, regarding how the parallel execution of three pipelines is useful for DSP related instructions**).

21.   Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Schlansker'538, in view of Matsushima (U.S. Patent No. 5,208,781), herein referred to as Matsushima'781.

22.   As to claim 10, Schlansker'538 discloses the claimed invention except a data

processing apparatus according to Claim 9, wherein the particular one of the memory

units is arranged to switch to a power saving state when the modified instruction

address is outside an address range or set of address ranges that contains said first

number of instruction addresses.

However, Matsushima'781 discloses where the particular one of the memory

units is arranged to switch to a power saving state when the modified instruction

address is outside an address range or set of address ranges that contains said first

number of instruction addresses (**see col. 3, lines 15-26, regarding switching to a**

power saving state if the address decoder outputs an address outside the range

of addresses of a memory chip).

Schlansker'538 and Matsushima'781 are analogous art because they are both

attempting to solve the problem of using memory more efficiently in a system (**see**

Schlansker'538, col. 2, lines 11-13, regarding the problem of using too much

memory in standard VLIW programs. See Matsushima'781, col. 3, lines 15-26,

regarding conserving power for memory chips).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Schlansker'538 to comprise where the particular one of

the memory units is arranged to switch to a power saving state when the modified

instruction address is outside an address range or set of address ranges that contains

said first number of instruction addresses, as taught by Matsushima'781, in order to

conserve power when the memory unit isn't being used (**see col. 3, lines 15-26,**

regarding the purpose being to save power which would be consumed by the memory chip).

CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

23.   The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

24.   Per the instant office action, claims 1-14 have received a first action on the merits and are subject of a first action non-final.

b. DIRECTION OF FUTURE CORRESPONDENCES

25.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alan Otto whose telephone number is 571-270-1626. The examiner can normally be reached on 8:00-5:30 M-F.

IMPORTANT NOTE

26.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on 571-272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

27.   Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AO

AO

Mano Padmanabhan

TC 2100

MANO PADMANABHAN

Don Sparks.